Information-theoretic inflectional classification

Sacha Beniamine[1,2], Olivier Bonami[1], Benoît Sagot[2]
LLF[1], Alpage[2], Université Paris Diderot

First International Quantitative Morphology Meeting (IQMM1)
University of Belgrade, 11 July – 15 July 2015

- Concept of Inflection Classes widely used to analyse inflectional systems

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :
  - Formal definitions of the concept

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :
  - Formal definitions of the concept
  - Large datasets

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :
    - Formal definitions of the concept
    - Large datasets
    - Reproducible classifications

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :
  - Formal definitions of the concept
  - Large datasets
  - Reproducible classifications
  - Commensurable in several languages

- Concept of Inflection Classes widely used to analyse inflectional systems
- **No consensus** on how to obtain the classification
- **Exploration of the concept through computational means** :
  - Formal definitions of the concept
  - Large datasets
  - Reproducible classifications
  - Commensurable in several languages
  - Basis for theoretical and typological comparisons.

Groups of lexemes that inflect alike.

	INF	PRES.3.SG	PRES.3.PL	PP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

Groups of lexemes that inflect alike .

	INF	PRES.3.SG	PRES.3.PL	PP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

1. What form should an IC system take ?
2. What generalisations should we infer from the data ?
3. How do we measure which lexemes inflect alike ?
4. How do we find the best classes among all possible ones ?

Table of Contents

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :
  - ***Cohesive**: Maximal homogeneity within classes*

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :
  - ***Cohesive***: *Maximal homogeneity within classes*
  - ***Distinctive***: *Maximal heterogeneity between classes*

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :
  - ***Cohesive***: *Maximal homogeneity within classes*
  - ***Distinctive***: *Maximal heterogeneity between classes*
- In most languages, the two criteria lead to different partitions :

Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

Inflection classes: Cohesive or distinctive ?

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :
  - ***Cohesive***: *Maximal homogeneity within classes*
  - ***Distinctive***: *Maximal heterogeneity between classes*
- In most languages, the two criteria lead to different partitions :
  - **Perfectly cohesive** : *numerous small, similar classes*

Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

- Insight from canonical typology (Corbett, 2009).
  An inflection class system is a partition that is :
  - ***Cohesive***: *Maximal homogeneity within classes*
  - ***Distinctive***: *Maximal heterogeneity between classes*
- In most languages, the two criteria lead to different partitions :
  - **Perfectly cohesive** : *numerous small, similar classes*
  - **Perfectly distinctive** : *fewer large classes with exceptions*

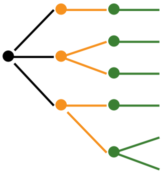
Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

Inflection classes: Cohesive or distinctive ?

- Dressler and Thornton's terminology (1996):
- Micro-classes
  - *Internally homogeneous : numerous small, similar classes*
- Macro-classes
  - *Externally heterogeneous : fewer large classes with exceptions*

Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

- Dressler and Thornton's terminology (1996):
- **Micro-classes**
  - *Internally homogeneous : numerous small, similar classes*
- **Macro-classes**
  - *Externally heterogeneous : fewer large classes with exceptions*
- Combined in a hierarchy. (Corbett and Fraser, 1993; Dressler and Thornton, 1996; Brown and Evans, 2012)



Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

Table of Contents

- **Stem and exponents** $/\text{bæg}/ + \left\{ \begin{array}{c} /z/(\textsc{pl}) \\ \emptyset\ (\textsc{sg}) \end{array} \right\}$
  - *Captures differences between cells under the assumption of a* ***constant stem****.*

- **Stem and exponents** $\quad /\text{bæg}/ + \begin{Bmatrix} /z/(\textsc{pl}) \\ \emptyset\ (\textsc{sg}) \end{Bmatrix}$
  - *Captures differences between cells under the assumption of a **constant stem**.*

- **Binary alternation patterns**

$$/\textit{bæg}/\ (\textsc{sg}) \rightleftharpoons /\textit{bægz}/\ (\textsc{pl})$$
$$/X/\ (\textsc{sg}) \rightleftharpoons /Xz/\ (\textsc{pl})$$

  - *Captures the **implicative relation** between each pair of cells.*

- **Stem and exponents** $\quad /\text{bæg}/ + \left\{ \begin{matrix} /z/ (\textsc{pl}) \\ \emptyset\ (\textsc{sg}) \end{matrix} \right\}$
  - *Captures differences between cells under the assumption of a* ***constant stem****.*
- **Binary alternation patterns** $\quad \begin{matrix} /bæg/\ (\textsc{sg}) \rightleftharpoons /bægz/\ (\textsc{pl}) \\ /X/\ (\textsc{sg}) \rightleftharpoons /Xz/\ (\textsc{pl}) \end{matrix}$
  - *Captures the* ***implicative relation*** *between each pair of cells.*
- Crucial difference:
  - global segmentation over the whole paradigm.
  - local segmentation over pairs of forms.

- **Global** : On the basis of a whole paradigm.
- **Local** : On each pair of cells.

Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	təniʁ	tjɛ̃	tjɛn	təny
FINIR 'finish'	finiʁ	fini	finis	fini
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	pəle	pɛl	pɛl	pəle
LAVER 'wash'	lave	lav	lav	lave
TASSER 'press'	tase	tas	tas	tase

- **Global** : On the basis of a whole paradigm.
- **Local** : On each pair of cells.

Lexeme	INF	PRS.3SG	PRS.3PL	PST.PTCP
TENIR 'hold'	Xəniʁ	Xjɛ̃	Xjɛn	Xəny
FINIR 'finish'	Xʁ	X	Xs	X
HAÏR 'hate'	aiʁ	ɛ	ais	ai
PELER 'peel'	X_1əX_2e	X_1ɛX_2	X_1ɛX_2	X_1əX_2e
LAVER 'wash'	Xe	X	X	Xe
TASSER 'press'	Xe	X	X	Xe

- **Global** : On the basis of a whole paradigm.
- **Local** : On each pair of cells.

Lexeme	INF ⇌ PRS.3SG	INF ⇌ PRS.3PL	INF ⇌ PST.PTCP	…
TENIR 'hold'	Xəniʁ ⇌ Xjɛ̃	Yəniʁ ⇌ Yjɛn	Ziʁ ⇌ Zy	
FINIR 'finish'	Xʁ ⇌ X	Yʁ ⇌ Ys	Zʁ ⇌ Z	
HAÏR 'hate'	aiʁ ⇌ ɛ	Yʁ ⇌ Ys	Zʁ ⇌ Z	…
PELER 'peel'	X_1əX_2e ⇌ X_1ɛX_2	Y_1əY_2e ⇌ Y_1ɛY_2	Z ⇌ Z	
LAVER 'wash'	Xe ⇌ X	Ye ⇌ Y	Z ⇌ Z	
TASSER 'press'	Xe ⇌ X	Ye ⇌ Y	Z ⇌ Z	

- This leads us to a **clustering problem** :
  - Evaluate fitness of clusters
  - Explore the search space

- **Minimum description length** (Rissanen, 1984) : Choose the model allowing for the shortest description of the data.
- A partition of the set of lexemes is better than another one if it leads to a more economical description of the system. (Sagot and Walther, 2011; Walther, 2013)

$$DL(description) =$$

$$length(description) \times - \sum_{x \in symbols(description)} P(x) log_2(P(x))$$

- **Microclasses**: number of bits necessary to encode the mapping between each lexeme and its microclass.

$$M = - \sum_{m \in Microclasses} |m| \times \log_2\left(\frac{|m|}{|lexemes|}\right)$$

- **Microclasses**: number of bits necessary to encode the mapping between each lexeme and its microclass.
- **Lexicon**: number of bits necessary to encode the mapping between each microclasse and its clusters.

$$L = -\sum_{c \in clusters} |c| \times \log_2\left(\frac{|c|}{|microclasses|}\right)$$

- **Microclasses**: number of bits necessary to encode the mapping between each lexeme and its microclass.
- **Lexicon**: number of bits necessary to encode the mapping between each microclasse and its clusters.
- **Grammar**: number of bits necessary to define the list of patterns available for each cluster and each pair of cells.

$$G = -\sum_{c \in clusters} \sum_{A,B \in pairs(cells)} |patterns_c^{A,B}| \times log_2(\frac{|patterns_c^{A,B}|}{|patterns^{A,B}|})$$

- **Microclasses**: number of bits necessary to encode the mapping between each lexeme and its microclass.
- **Lexicon**: number of bits necessary to encode the mapping between each microclasse and its clusters.
- **Grammar**: number of bits necessary to define the list of patterns available for each cluster and each pair of cells.
- **Residual ambiguity**: number of bits necessary to encode the mapping of each microclass to its pattern given the lexicon and the grammar

$$R = -\sum_{c \in clusters} \sum_{A,B \in pairs(cells)} \sum_{t \in patterns_c^{A,B}} |t| \times \log_2(\frac{|t|}{|c|})$$

- **Microclasses**: number of bits necessary to encode the mapping between each lexeme and its microclass.
- **Lexicon**: number of bits necessary to encode the mapping between each microclasse and its clusters.
- **Grammar**: number of bits necessary to define the list of patterns available for each cluster and each pair of cells.
- **Residual ambiguity**: number of bits necessary to encode the mapping of each microclass to its pattern given the lexicon and the grammar

$$DL = M + L + G + R$$

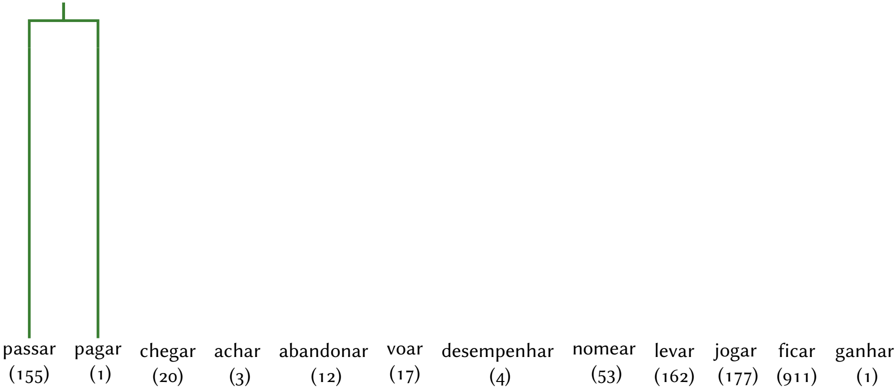
Table of Contents

(a) Merge lexemes with identical patterns (micro-classes).

passar	pagar	chegar	achar	abandonar	voar	desempenhar	nomear	levar	jogar	ficar	ganhar
(155)	(1)	(20)	(3)	(12)	(17)	(4)	(53)	(162)	(177)	(911)	(1)

Clustering algorithm

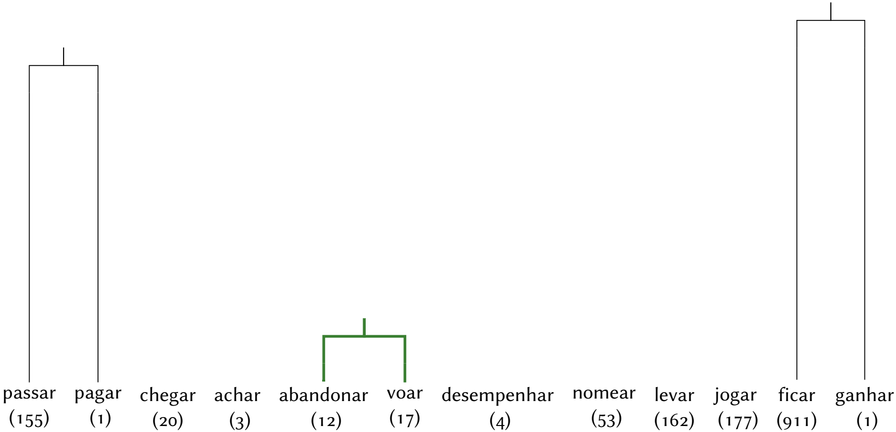
(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).



Clustering algorithm

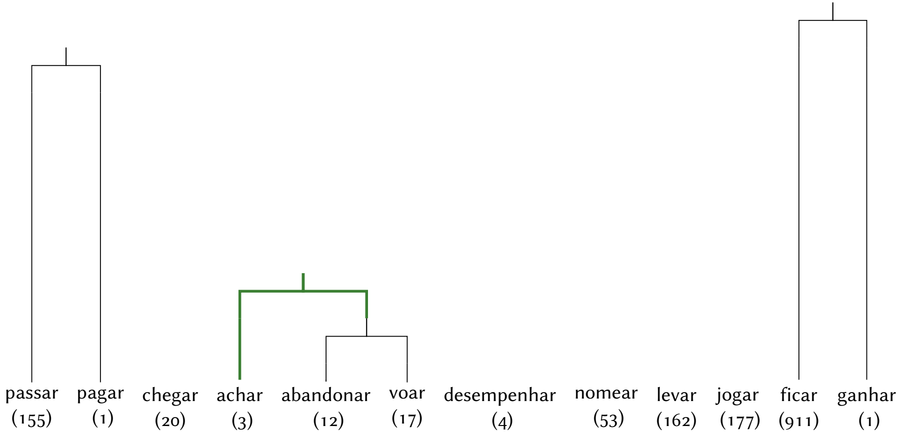
(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.



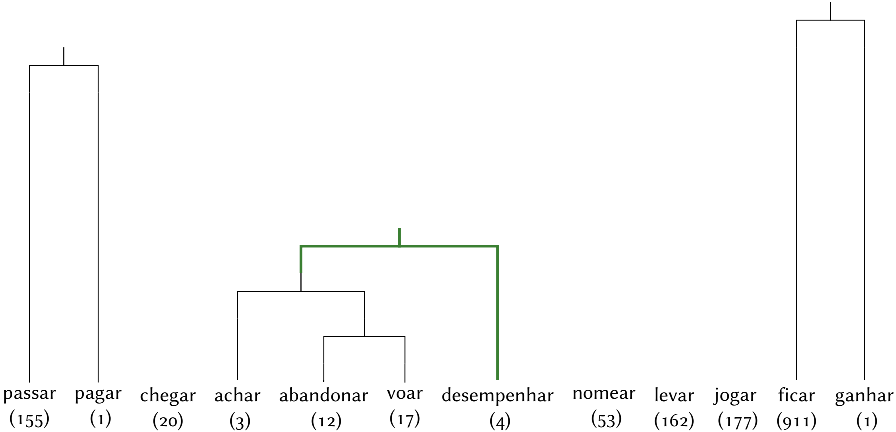
Clustering algorithm

(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

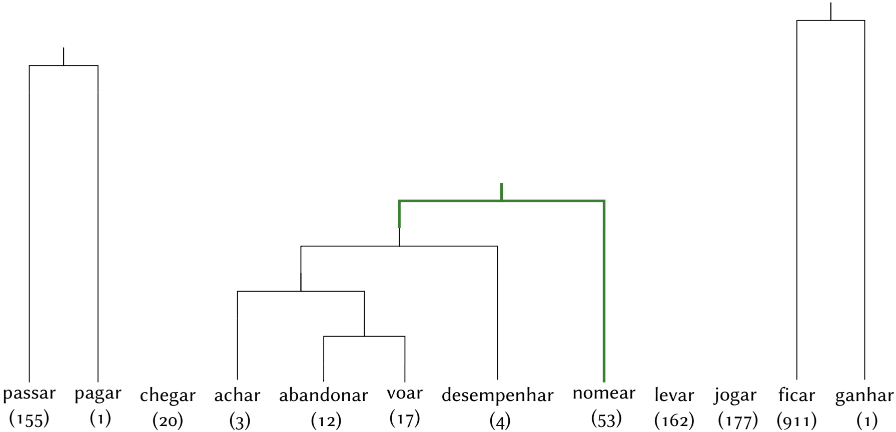


Clustering algorithm

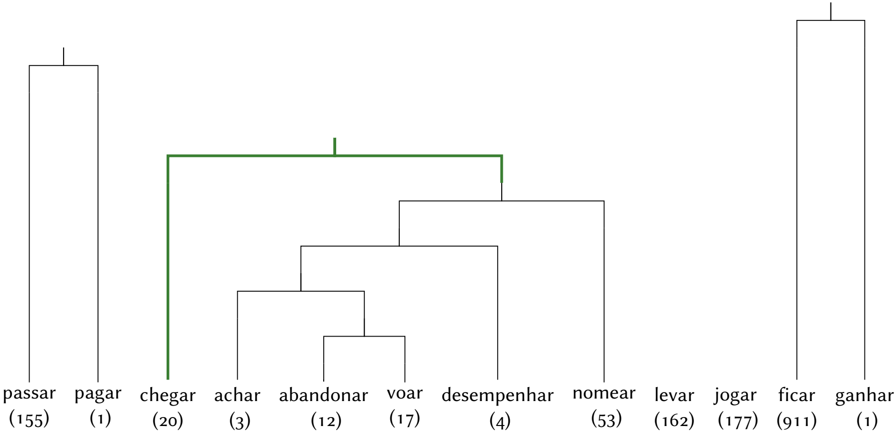
(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.



(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

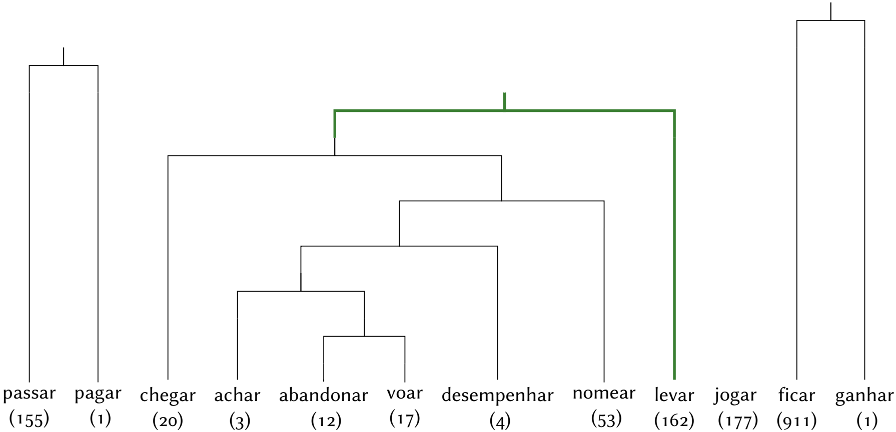


(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

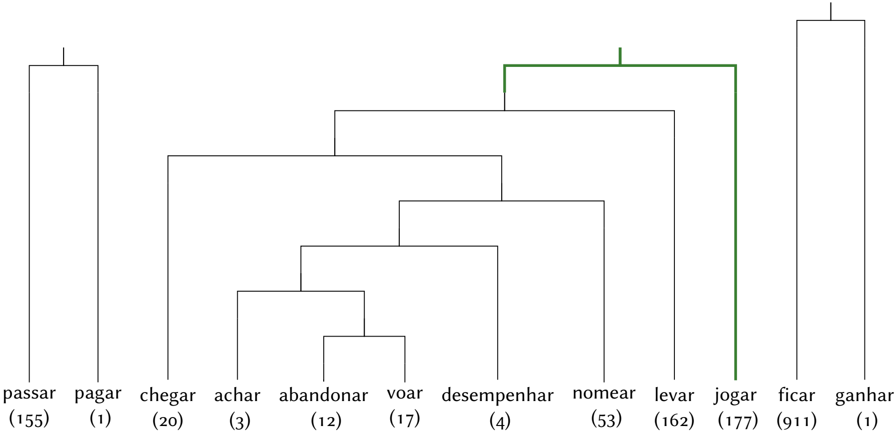


Clustering algorithm

(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

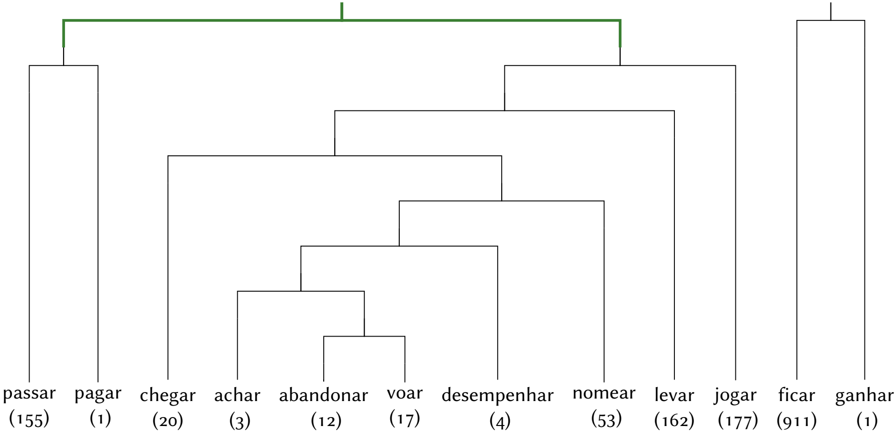


(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.



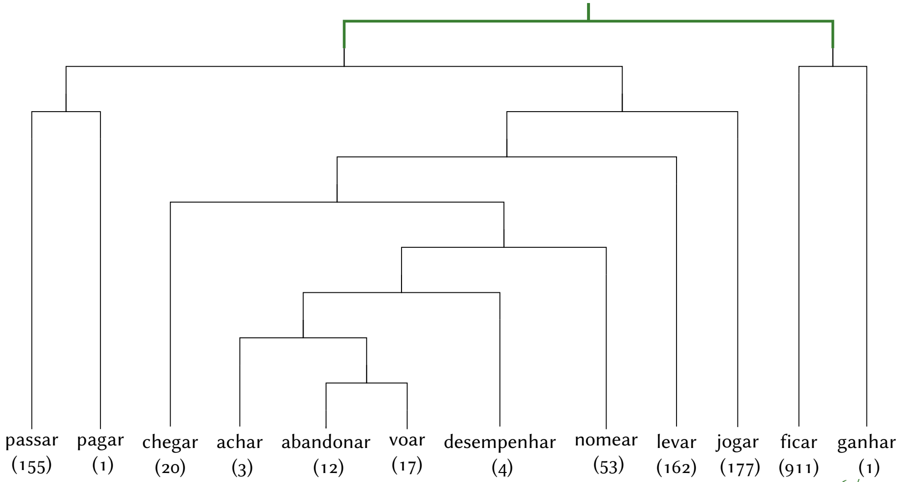
Clustering algorithm

(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.

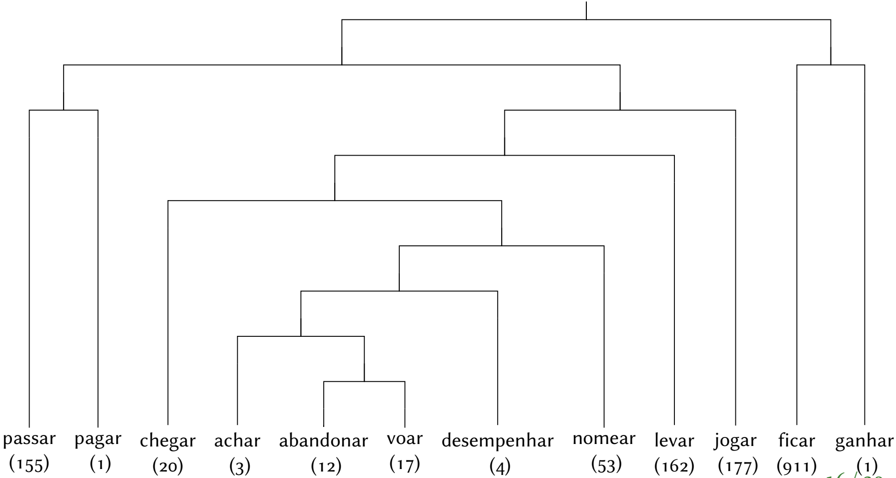


Clustering algorithm

(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.



(a) Merge lexemes with identical patterns (micro-classes).
(b) Merge the pair optimising descriptive economy (lowest DL).
(c) Repeat until there is only 1 cluster.



- As we merge clusters, we expect the DL to decrease.
- We define **macroclasses** when at some point, merging doesn't lead to a decrease.
  - Macroclasses optimise the size of the system, without any information loss.
  - It is an empirical issue wether a system has macroclasses or not.
- They do show up in our experiments.

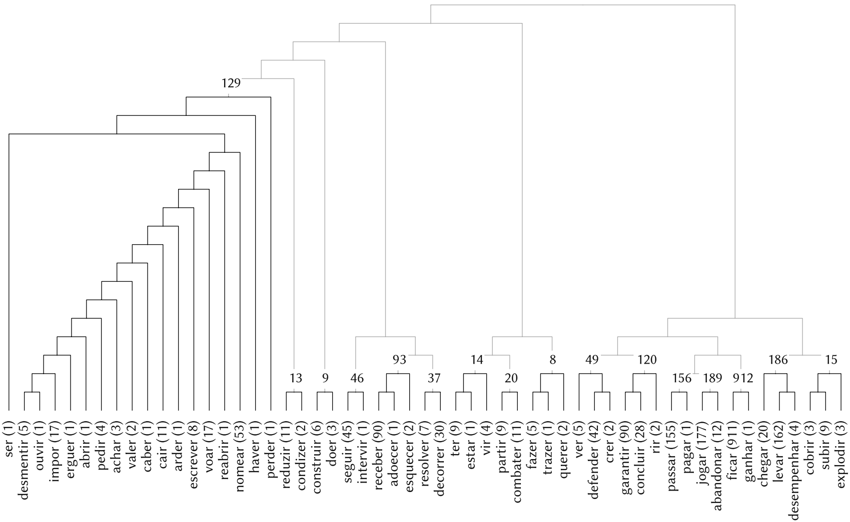
Comparison to other works

	Generalisations	Criterion	Algorithm
Brown and Evans (2012)	raw paradigms	Compression distance	CompLearn
Bonami (2014)	Affixes	Edit distance	UPGMA
Bonami (2014)	Patterns	Hamming distance	UPGMA
Lee and Goldsmith (2013)	Sets of characters	DL variant	greedy bottom-up
Here	Local patterns	DL	greedy bottom-up
Here	Global patterns	DL	greedy bottom-up

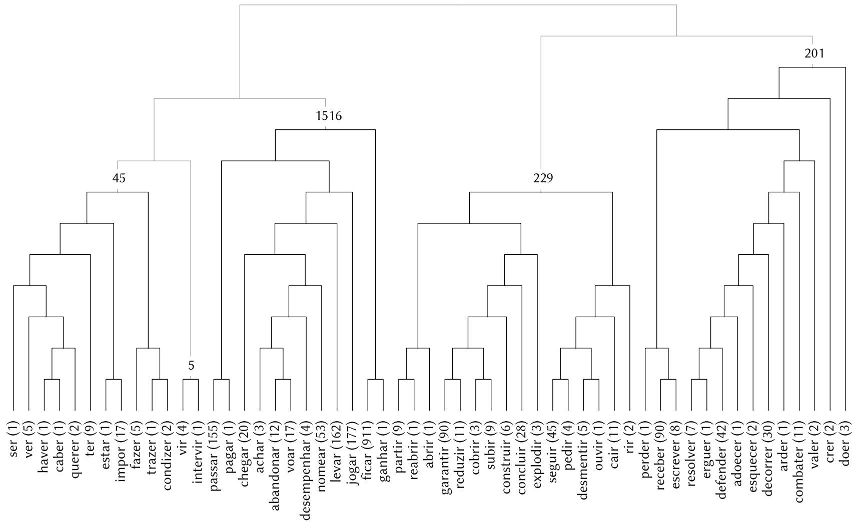
Table of Contents

- Paradigm tables in phonemic transcription.
- **European Portugese**: Coimbra pronunciation dictionnary (Veiga et al., 2013) (1995 verbal entries).
- **French**: Flexique (Bonami et al., 2014) (5406 verbal entries).

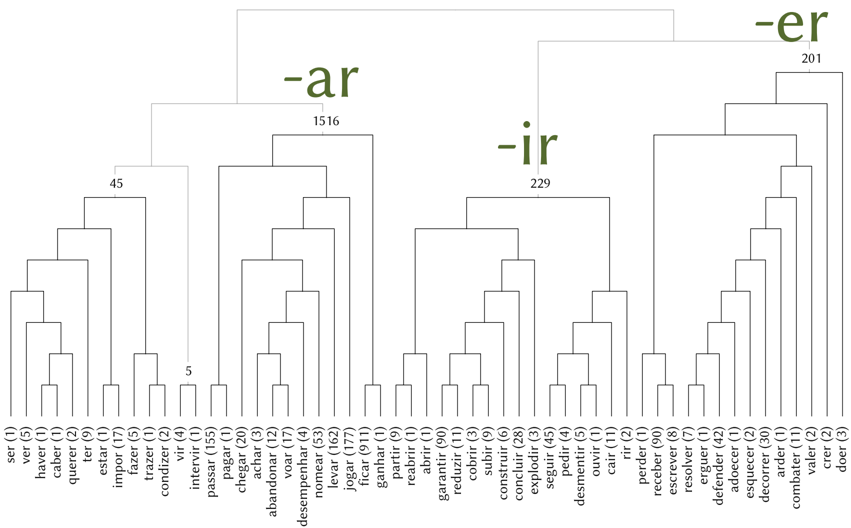
Portugese inflection classes, with global templates



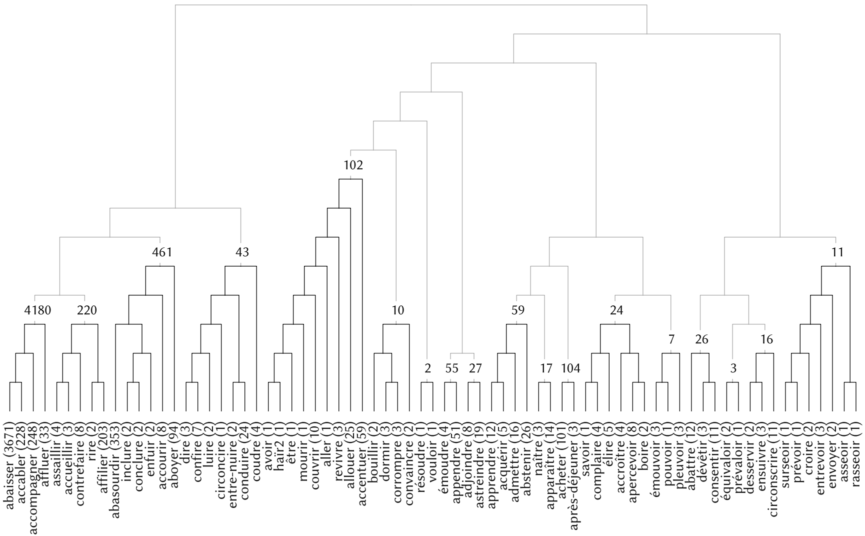
Portugese inflection classes, with local templates



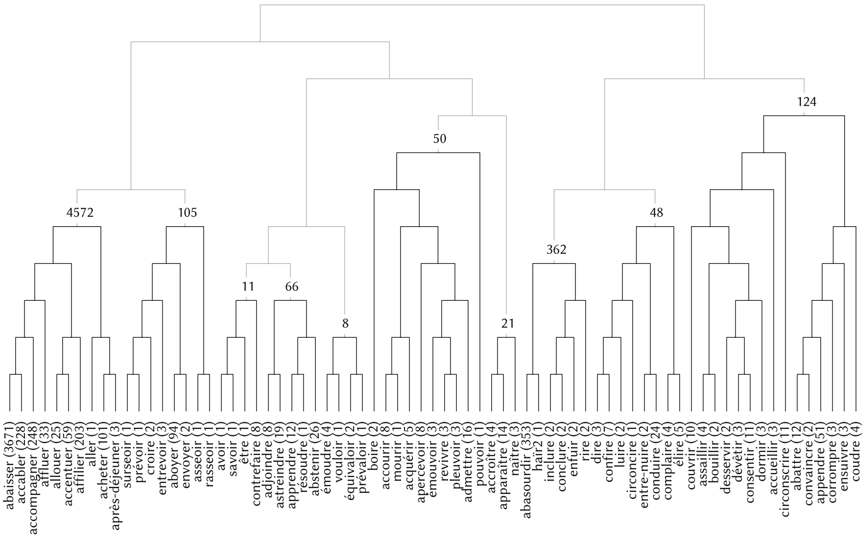
Portugese inflection classes, with local templates



French inflection classes, with global templates



French inflection classes, with local templates



- **Main properties** :
  - Based on information-theoretic measures.
  - Relies on automatically inferred generalisations.
- **Compared to previous attempts** to automatise IC :
  - Segmentations not taken as granted.
  - Using a measure that evaluates the quality of the system allows us to infer relevant macroscopic generalisations.
  - No parameters to adjust: **Occam's razor** is the only criterion.
- **Local segmentation** better captures the structure in inflection system than global segmentations.
  - Supports the relevance of local patterns of alternation in abstractive approaches (Blevins, 2006).
  - Complementary to work on information-theoretic modeling of implicative structure (Ackerman et al., 2009; Ackerman and Malouf, 2013; Bonami and Beniamine, 2015)
- **Freely available code** : `http://qumin.gforge.inria.fr`

Qumin

Ackerman, F., Blevins, J. P., and Malouf, R. 2009. Parts and wholes: Patterns of relatedness in complex morphological systems and why they matter.

Ackerman, F. and Malouf, R. 2013. Morphological organization: The low conditional entropy conjecture. *Language* 89:429–464.

Blevins, J. P. 2006. Word-based morphology. *Journal of Linguistics* 42:531–573.

Bonami, O. 2014. La structure fine des paradigmes de flexion. Habilitation à diriger des recherches, U. Paris Diderot.

Bonami, O. and Beniamine, S. 2015. Implicative structure and joint predictiveness.

Bonami, O., Caron, G., and Plancq, C. 2014. Construction d'un lexique flexionnel phonétisé libre du français. *In* Actes du quatrième Congrès Mondial de Linguistique Française, pp. 2583–2596.

References II

Brown, D. and Evans, R. 2012. Morphological complexity and unsupervised learning: validating Russian inflectional classes using high frequency data, pp. 135–162. *In* F. Kiefer, M. Ladányi, and P. Siptár (eds.), Current Issues in Morphological Theory: (Ir)regularity, analogy and frequency. John Benjamins, Amsterdam.

Corbett, G. G. 2009. Canonical inflectional classes. *In* Selected Proceedings of the 6th Décembrettes: Morphology in Bordeaux.

Corbett, G. G. and Fraser, N. M. 1993. Network morphology: a DATR account of Russian nominal inflection. *Journal of Linguistics* 29:113–142.

Dressler, W. U. and Thornton, A. M. 1996. Italian nominal inflection. *Wiener Linguistische Gazette* 55-57:1–26.

Lee, J. and Goldsmith, J. A. 2013. Automatic morphological alignment and clustering. Presented at the 2nd American International Morphology Meeting.

Rissanen, J. 1984. Universal coding, information, prediction, and estimation. *IEEE Tr. on Info. Th.* 30:629–636.

Sagot, B. and Walther, G. 2011. Non-canonical inflection: data, formalisation and complexity measures. *In* C. Mahlow and M. Piotrowski (eds.), Systems and Frameworks in Computational Morphology, volume 100 of *Communications in Computer and Information Science*, pp. 23–45, Zurich, Suisse. Springer.

Veiga, A., Candeias, S., and Perdigão, F. 2013. Generating a pronunciation dictionary for european portuguese using a joint-sequence model with embedded stress assignment. *Journal of the Brazilian Computer Society* 19:127–134.

Walther, G. 2013. On canonicity in morphology:an empirical, formal and computational approach. PhD thesis, Université Paris Diderot, École doctorale de sciences du langage 132, U.F.R. de linguistique.